AD_______________

GRANT NO: DAMD17-94-J-4328

TITLE: Computer-aided Diagnosis and Automated Screening of Digital Mammograms

PRINCIPAL INVESTIGATOR: Dr. Kevin S. Woods

CONTRACTING ORGANIZATION: University of South Florida
Tampa, FL 33620

REPORT DATE: Sept. 29, 1995

TYPE OF REPORT: Annual

PREPARED FOR: U.S. Army Medical Research and Materiel
Command
Fort Detrick, Maryland 21702-5012

DISTRIBUTION STATEMENT: Approved for public release;
distribution unlimited

The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision unless so designated by other documentation.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
Sept. 29, 1995

3. REPORT TYPE AND DATES COVERED
Annual 1 Sep 94 - 31 Aug 95

4. TITLE AND SUBTITLE
Computer-Aided Diagnosis and Automated Screening of Digital Mammogram

5. FUNDING NUMBERS
DAMD17-94-J-4328

6. AUTHOR(S)
Dr. Kevin S. Woods

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of South Florida
Tampa, Florida 33620

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Medical Research and Materiel Command
Fort Detrick, Maryland 21702-5012

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The primary objective of the proposed research is to develop computer techniques, methods and rigorous testing to provide reliable results which can be used by radiologists to increase diagnostic accuracy and decrease the rate of undetected cancer First, "basic" level computer science algorithms and concepts have been developed and refined. These research topics include development of a novel algorithm for generating ROC curves for artificial neural networks, a method for multiple classifiers. These algorithms are fundamental building blocks to be used in the construction of the final system. The second general area of research is more application-specific. The topic of this research has been to develop reliable routines for segmenting mammographic abnormalities using texture features, as opposed to contrast or pixel intensity features.

14. SUBJECT TERMS
Digital Mammography, ROC Analysis, Combination of Multiple Classifiers, Tumor Segmentation breast cancer

15. NUMBER OF PAGES
38

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	Unlimited

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements***.

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	- Contract	**PR**	- Project
G	- Grant	**TA**	- Task
PE	- Program Element	**WU**	- Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE - See authorities.
NASA - See Handbook NHB 2200.2.
NTIS - Leave blank.

Block 12b. Distribution Code.

DOD - Leave blank.
DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
NASA - Leave blank.
NTIS - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

★U.S.GPO:1991-0-305-776

FOREWORD

Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the US Army.

____ Where copyrighted material is quoted, permission has been obtained to use such material.

____ Where material from documents designated for limited distribution is quoted, permission has been obtained to use the material.

____ Citations of commercial organizations and trade names in this report do not constitute an official Department of Army endorsement or approval of the products or services of these organizations.

____ In conducting research using animals, the investigator(s) adhered to the "Guide for the Care and Use of Laboratory Animals," prepared by the Committee on Care and Use of Laboratory Animals of the Institute of Laboratory Resources, National Research Council (NIH Publication No. 86-23, Revised 1985).

____ For the protection of human subjects, the investigator(s) adhered to policies of applicable Federal Law 45 CFR 46.

____ In conducting research utilizing recombinant DNA technology, the investigator(s) adhered to current guidelines promulgated by the National Institutes of Health.

____ In the conduct of research utilizing recombinant DNA, the investigator(s) adhered to the NIH Guidelines for Research Involving Recombinant DNA Molecules.

____ In the conduct of research involving hazardous organisms, the investigator(s) adhered to the CDC-NIH Guide for Biosafety in Microbiological and Biomedical Laboratories.

Kevin Woods 9/29/95
PI - Signature /Date

Table of Contents

1 Introduction

The primary objective of the proposed research is to develop computer techniques for detecting abnormalities in digital mammograms using sound engineering methods and rigorous testing to provide reliable results which can be used by radiologists to increase diagnostic accuracy and decrease the rate of undetected cancers. To provide flexibility in a clinical setting, a radiologist will have control over the sensitivity/specificity of computer detection routines. In order to provide this flexibility, receiver operator characteristic (ROC) techniques will be developed for non-traditional statistical classifiers, such as artificial neural networks (ANNs) and binary decision trees (BDTs). Finally, all software will be optimized and packaged into a graphical user interface which will provide a radiologist with access to image processing tools in addition to the computer detection routines.

The first full year of research in computer-aided diagnosis of mammogram images can be divided into two general areas. First, "basic" level computer science algorithms and concepts have been developed and refined. During the process of developing solutions for the digital mammography application, problems of a more basic nature arise. These basic problems usually require general purpose solutions which, in turn, result in techniques that are applicable in many areas of computer science and engineering. Second, application specific experiments have been performed which are directed towards realization of the primary objective of the funded research. Namely, to develop a fully automated system for computer-aided diagnosis of digital mammograms.

The following subsections will introduce the topics that have been the focus of the first full year of research. Here, the problems will be summarized, our solutions will be introduced, and some background information will be provided. The body of this report (Section 2) will provide the details of the research conducted, including experimental methods, data, and results. Section 3 of the report will summarize the research, and draw some conclusions. Each research topic will be covered in separate subsections within each of the three major sections of this report. Much of the following material has or will be submitted to scientific journals and/or conference proceedings.

1.1 ROC Analysis

The accuracy of a classifier (in a 2-class problem) can be completely characterized by a plot of the classifier's true positive detection rate versus its false positive rate. This is called a receiver operating characteristic (ROC) curve. The false positive (FP) rate is the probability of incorrectly classifying a normal sample as an abnormal sample. The true positive (TP) detection rate, also called the sensitivity, is the probability of correctly classifying an abnormal sample. The TP and FP rates both are specified in the interval from 0.0 to 1.0 (or 0% to 100%), inclusive. Statistical classifiers have parameters that can be varied to alter the TP and FP rates. Each set of parameter values may result in a different (TP,FP) pair, called an operating point.

The Area Under the ROC Curve (AUC) is an accepted way of comparing *overall* classifier performance [1, 2, 3]. However, the "best" classifier for an application may well depend on the particular combination of TP and FP rates that are required. Additionally, classifier performance at very low sensitivities or very high FP rates is usually not of practical interest.

When large portions of the ROC curve lie outside the range of interest, it is more useful to analyze only a portion of the curve.

The objective of this research subtopic was to review current methods for generating ROC curves for traditional and non-traditional classifiers, and to develop scientifically sound techniques for comparing ROC curves generated by different classifiers. Generating ROC curves for traditional classifiers, such as Bayesian and K-Nearest Neighbor classifiers, are well established. Similarly, techniques for comparing full and partial ROC curves have been explored in previous work [2, 4]. However, since ROC analysis is so fundamentally crucial to the proposed research, a review of established techniques was essential. Upon completion of this review, we found it was necessary to refine some existing techniques for comparing portions of ROC curves.

1.2 Generating ROC Curves for ANNs

While techniques for generating ROC curves for "traditional" classifiers are well understood, the same cannot be said for "non-traditional" classifiers such as ANNs and decision trees. In a majority of previous work, artificial neural networks (ANNs) have been applied as a classifier to find one "best" detection rate. Recently researchers have begun to report ROC curve results for ANN classifiers. The current standard method of generating a ROC curve for an ANN is to vary the output node threshold for classification. We developed a novel algorithm for generating a ROC curve for a 2-class ANN classifier. By appropriately scaling the bias input weight for selected nodes on the first hidden layer, we can control the ANN's TP and FP rates in a desirable manner. The training data is used to determine the sets of scale factors that will change the TP rate from 0% to 100% while maintaining as low a FP rate as possible. We show that this new technique generates better ROC curves in the sense of having greater area under the ROC curve, and in the sense of being composed of a greater number of distinct operating points. As a result of applying our algorithm, the same ANN is capable of achieving not only better detection rates, but also permits the user to select an appropriate operating point for a given application.

1.3 Combination of Multiple Classifiers

Many pattern classification techniques exist, each with inherent strengths and weaknesses. No single classifier is suitable for all applications. Similarly, it may be difficult to achieve acceptable performance for complex data distributions using any single classifier. One alternative in the search for greater performance is integrated or adaptive methods which are capable of capitalizing on the strengths of several individual classifiers. Indeed, recent work in handwritten character recognition has shown promising results when several classifiers are combined to reach a decision [5, 6, 7, 8, 9, 10, 11, 12, 13]. This is often referred to as Combination of Multiple Classifiers (CMC). We have been exploring CMC in order to determine the merit of such approaches for improved classification accuracy in a digital mammography application.

Most of the current approaches to CMC treat the training data as a monolithic whole when determining classifier accuracy. However, it seems intuitive that the accuracy might vary with position in feature space. Given an arbitrary test sample and classifiers which may have different feature spaces, it is reasonable to think that a given classifier would perform similarly for other samples near the test sample in its feature space. The result of our CMC research is a novel algorithm for dynamic classifier selection that uses estimates of a classifier's accuracy in local regions of feature space. Our CMC algorithm simply selects the classifier which is most accurate for a subset of training samples nearest to the test sample. Note that the particular samples in the subset may vary among classifiers.

1.4 Segmentation Techniques

The research topics in the previous 3 Subsections have dealt with computer science topics that are general-purpose in nature. The research introduced in this subsection is concerned with determining the most promising *fundamental* approach to segmenting/detecting abnormalities in mammogram images.

Most approaches to date have used some type of contrast-based segmentation scheme to extract potentially abnormal tumor regions from digital mammograms. Since most lesions are more radiographically dense compared to surrounding tissue, a contrast-based approach seems intuitively appealing. However, the vast range in appearance of typical lesions makes it very difficult to develop a reliable segmentation scheme with a high specificity. That is, in order for a segmentation algorithm which relies heavily on pixel intensity to segment a majority of lesions, a large number of false positive regions will also be segmented. The false positive rate of the contrast-based segmentation approaches is so high that even subsequent processing of the images still fails to raise the system specificity to an acceptable level.

An alternative to contrast-based segmentation is a texture-based segmentation scheme. This type of approach has been used successfully to detect spiculated lesions with an acceptable false positive rate [14, 15, 16]. Recent work comparing contrast-based segmentation to texture-based segmentation [17] would seem to indicate the importance of considering the texture of the breast tissue during the segmentation process. The goal of this research subtopic is to develop a general-purpose texture-based segmentation scheme that can be used to segment most kinds of mammographic abnormalities. The initial phases of this research have been concerned with determining what kinds of texture features are useful, and what kind of initial results can be expected using such an approach.

2 Body

This section provides details of the research directed towards solving the 4 problems introduced in the Section 1.1 through 1.4. Whenever possible, experimental methods and results are provided.

2.1 Generating and Comparing ROC Curves

The following 4 subsections (2.1.1 to 2.1.4) briefly review current techniques for generating ROC curves for several families of statistical classifiers. Subsection 2.1.5 presents the methods we use for comparing ROC curves generated by two different methods of classification. This work involved refining an existing test for statistical significance so that it was applicable to our problem domain.

2.1.1 ROC for Bayesian Classifiers

For the Bayesian (LC and QC) classifiers, we get as output the *a posteriori* probability of an unknown sample for both classes. We compute the ratio

$$r = \frac{P_{abn}}{P_{nrml}} \tag{1}$$

where P_{abn} and P_{nrml} are the *a posteriori* probabilities of the sample belonging to the abnormal class and the normal class, respectively. We can set a decision threshold, T, such that if the ratio is greater than T, the unknown sample is classified as abnormal, otherwise it is labeled as normal. By varying the threshold, the TP/FP trade-off of the Bayesian classifiers can be altered. For example, as T is increased, both the TP and FP rates will decrease (not necessarily at the same rate).

2.1.2 ROC for KNN Classifier

For the KNN algorithm, ROC points for a specific value of K are obtained by varying k (the number of votes required) for the abnormal class from 1 to K and observing the resulting TP and FP rates. It should be apparent that as k is increased, the TP and FP rates will decrease since it will require more "votes" for an object to be classified as abnormal. To optimize the KNN classifier, we vary the value of K from 1 to 200. At each value, an ROC curve is generated by varying k from 1 to K, and the AUC is computed. The K value that produces the maximum AUC is selected for classification purposes. The operating point of the KNN classifier can then be selected by choosing an appropriate value of k.

2.1.3 ROC for Decision Trees

In decision trees, the leaf nodes may be seen as associating a probability with each class. The probability is computed from the training samples that fall into the leaf after the tree has been grown and pruned. For example, a leaf node may contain 80 training samples from class 1, and 20 training samples from class 2. During classification, we can say an unknown sample that falls into this leaf has an 80% probability of belonging to class 1, or a 20% probability of belonging to class 2. Thus, ROC points for decision trees are obtained by simply varying a threshold for the probability of a sample belonging to the abnormal class. So, as the threshold for the abnormal class is lowered we would expect more samples to be classified as abnormal, thereby increasing both the TP and FP rates.

2.1.4 ROC for ANNs

In the Subsection 2.2, we describe, in detail, the traditional method of generating ROC curves for ANNs, and the novel algorithm that we developed in response to some drawbacks of the traditional method.

2.1.5 A Test for Statistical Significance

Hanley and McNeil [2] describe methods to determine if the observed difference between two AUCs is statistically significant. An AUC that has been computed over a full ROC curve is equivalent to the probability that a randomly selected abnormal sample will be rated more suspicious, by a classifier or a human, than a randomly selected normal sample [2]. An AUC computed over a portion of a ROC curve is equivalent to a conditional probability, and must be expressed as such prior to applying the methods of Hanley and McNeil.

First, the AUCs over the range of interest are estimated using the trapezoid rule for the discrete operating points. The area under a portion of a ROC curve can be expressed as a conditional probability via the following transformation:

$$AUC = \frac{A_p}{TP_2 - TP_1} \tag{2}$$

where A_p is the area under the ROC curve computed between TP rates TP_1 and TP_2. (A similar transformation would be used when AUCs are computed between FP rates FP_1 and FP_2.)

The formula for the z statistic is

$$z = \frac{AUC_1 - AUC_2}{\sqrt{{SE_1}^2 + {SE_2}^2}} \tag{3}$$

where AUC_1 and AUC_2 are the two estimated AUCs, and SE_1 and SE_2 are the estimated standard errors of each AUC. We use a two-tailed test for statistical significance. The null hypothesis is that the two observed AUCs are the same. The alternate hypothesis is that the two AUCs are different. A critical range of $z > 1.96$ or $z < -1.96$ (a level of significance $\alpha = 0.05$) indicates that the null hypothesis can be rejected, and there is sufficient evidence to support the alternate hypothesis.

A conservative estimate of the standard error of an AUC value can be calculated (from [2]) as:

$$SE(AUC_i) = \sqrt{\frac{\theta(1-\theta) + (n_A - 1)(Q_1 - \theta^2) + (n_N - 1)(Q_2 - \theta^2)}{n_A n_N}} \tag{4}$$

where Q_1 and Q_2 are two distribution-specific quantities, θ is the "true" area under the ROC curve, and n_A and n_N are the number of abnormal and normal samples, respectively. The estimate AUC_i is used as an estimate of θ. The quantities Q_1 and Q_2 are expressed as functions of θ:

$$Q_1 = \frac{\theta}{2 - \theta} \tag{5}$$

and

$$Q_2 = \frac{2\theta^2}{1 + \theta} \tag{6}$$

In a test for statistical significance, two ROC curves are compared *only* over the range of TP rates that are common to both curves.

2.2 Generating ROC Curves for ANNs

For the following discussions, it is assumed that the reader is familiar with ANN concepts in general, including activation functions, the role of the bias unit and the weights on the bias input to a node, and the backpropagation training algorithm. For a basic introduction to backpropagation neural nets, the reader is referred to [18].

2.2.1 The ANN architecture

All of the experiments reported here use fully-connected backpropagation networks with sigmoid activation functions (output range 0.0 to 1.0), a bias unit with weighted connections to all nodes, and a single output node. The value produced by the output node would typically be thresholded so that values greater than or equal to 0.5 are labeled "target", and values less than 0.5 are labeled "non-target". Our experiments utilize simulated and real data. We have created several sets of 2, 3, and 5-dimensional data. These simulated data sets will be described in more detail later. Some of the experiments on the simulated data utilize a network with 2 hidden layers with 10 hidden nodes per hidden layer and one output node. The number of inputs, obviously, depends on the dimensionality of the data set (i.e. either 2, 3, or 5 inputs). We also examine networks with a single hidden layer of 10 nodes and, as before, 1 output and either 2, 3, or 5 inputs. In our own previous work on a problem in mammogram image analysis [19], a network with 7 inputs (plus 1 bias input), 2 hidden layers with 10 hidden nodes per hidden layer, and 2 output nodes was found empirically to give the best performance. For consistency, this same architecture is used in this paper for all experiments that utilize the mammography dataset with the exception that only one output node is used.

2.2.2 ANN training and testing

The ANN must be trained before the ROC curve can be generated. The ANN is trained for 2000 epochs using a standard backpropagation learning algorithm, and the network weights are saved for the lowest network error (sum of squared error over all training samples) found during training. The resulting network is referred to as a "basic trained network". Each of the two methods we discuss manipulates one or more parameters of the basic trained network to generate a ROC curve. The current accepted method [20, 21, 22, 23], varies the threshold on the value produced by the output node. Our proposed new method scales the weight on the bias inputs for selected nodes on the first hidden layer.

It is important to follow careful methodology in selecting and evaluating ROC points. In a given experiment, the training data set is used to train the ANN. This fixes values for all of the weights in the ANN. This initial instance of the ANN provides one operating point. Based on the training data, values of some underlying network parameter(s) are selected to give additional instances of the ANN. The result is a set of instances of the network chosen to represent points on the ROC curve. The goodness of this set of network instances is then evaluated using a separate test set of data. So, ANN training involves both the learning of the network connection weights, and the estimation of classifier parameter settings for the purpose of generating a ROC curve.

The results for a trained ANN are naturally dependent on both the particular training set and the initial values of the weights. Therefore, it is necessary to observe a number of instances of a basic trained network in order to give a true picture of the relative performance of a method for generating ROC points. Thus, experimental results are obtained for several random initializations of the ANN weights for each set of training data. The training and test sets for each experiment are described in detail in the following subsection.

Occasionally, an ANN is unable to converge on a reasonable solution for the training data within the allotted 2000 training epochs. When this occurs, all training samples are assigned to the same class. This corresponds to a either a ROC point with 100% TP *and* FP rates, or a point with 0% TP *and* FP rates. A network instance which has trained to one of these extreme ROC points is effectively useless, and we do not attempt to extract ROC curves in this situation. We will note the number of times the ANN did not properly train in our experimental results.

2.2.3 The experimental data

We report experimental results for a number of different sets of simulated data. Simulated data is attractive from an experimental standpoint because we can create a virtually unlimited supply of completely independent training and test data with known distributions. Due to the limited number of target training samples available from our mammography data, each of the training sets will contain many of the same target samples. Therefore, while the training data is completely independent from the test data, the individual training sets are not independent of each other. The addition of the simulated data permits us to be reasonably sure that our test results are not somehow dependent on characteristics inherent in the mammography data, such as the number, dimensionality, and/or distribution of the data samples.

In our first sets of simulated data, samples from both the target class and non-target class have Gaussian distributions with some regions of overlap in feature space. We use 4 different size training sets (500, 1000, 1500, and 2000 samples) with an equal number of samples from the target and nontarget classes. We use data sets with 3 different input dimensions (2-d, 3-d, and 5-d). The target samples have a mean of 0.625 with a standard deviation of 0.1 for all dimensions, and the non-target samples have a mean of 0.375 with a standard deviation of 0.1 for all dimensions. Figure 1 shows an example of some 2-dimensional data with these distributions. We created 3 sets of test data (one for each different input dimension) of 5000 samples (half from each class) to test these first sets of simulated data. Overall, we have 12 sets of training data (4 $sizes \times 3$ $dimensions$) and 3 sets of test data of overlapping normally distributed data.

In our second sets of simulated data, only the target samples have a Gaussian distribution. The non-target samples have a uniform distribution. The non-target samples are permitted to overlap the target samples for some regions in feature space, but not near the mean of the target samples. As before, we use 4 different size training sets and 3 different input dimensions with half the samples coming from each class. The target samples have a mean of 0.75 with a standard deviation of 0.1 for all dimensions. Figure 2 shows an example of some 2-dimensional data with the distributions we have just described. Thus, we have another 12 sets of training data (4 $sizes \times 3$ $dimensions$). Again, we have 3 sets of test data

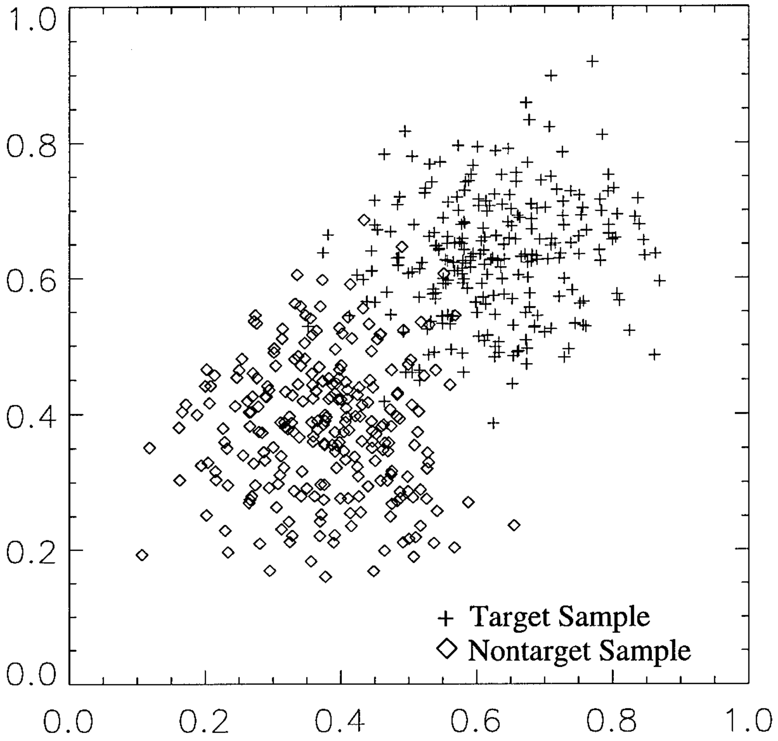


Figure 1: A portion of one of the training sets of 2-dimensional data in which both classes have Gaussian distributions.

(one for each different input dimension) with 5000 samples.

In addition to the simulated data, experimental results are reported for real data from a classification problem in mammogram image analysis. The data is a set of 2-class, 7-dimensional, normalized feature vectors that have been extracted automatically from labeled (ground truth), segmented mammogram images. The mammogram images, the feature selection and extraction methods, and the segmentation procedure are presented in [19]. The mammogram images are divided into *separate* sets of training and test images from which the training and test samples are obtained, respectively. Fifteen different training sets are created by randomly selecting an equal number of target and non-target samples from the set of training images. More specifically, we have 5 training sets with 300 samples, 5 sets with 400 samples, and 5 sets with 524 samples[1]. The mammography data test set includes all samples from all of the test images (280 targets, and 3719 nontargets).

For each of the 24 training sets of simulated data, we perform 3 random initializations of the ANN weights prior to training the network. As we mentioned before, simulated data will be used to train networks with 2 different architectures, one with a single hidden

[1] The set of training images contained a total of 262 target samples. Thus, the training sets with 524 samples contain all target samples available in the training images

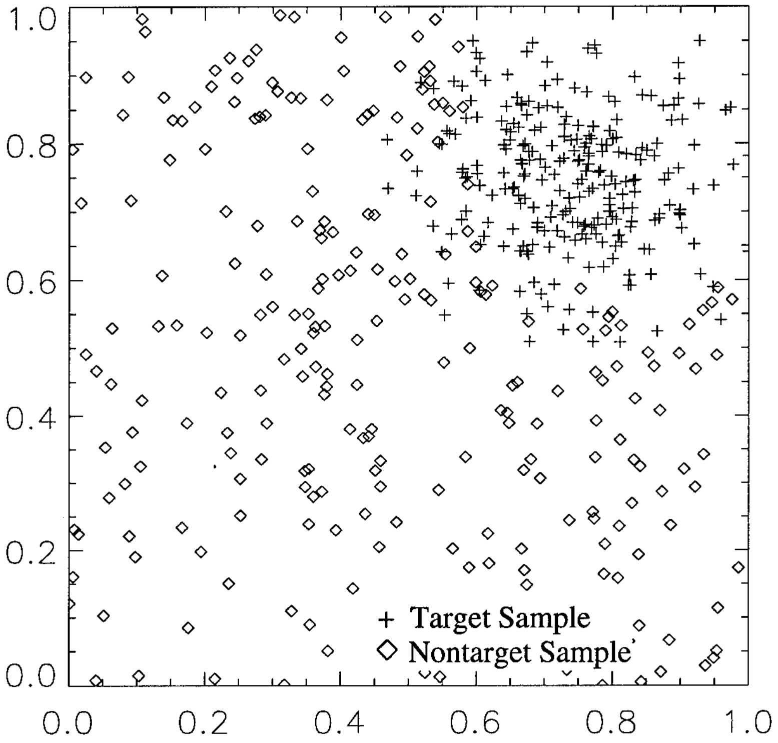


Figure 2: A portion of one of the training sets of 2-dimensional data in which only the target class has a Gaussian distribution, and the nontarget class is more uniformly distributed.

layer of 10 nodes, and one with two hidden layers of 10 nodes each. Therefore, we may have up to 144 instances of a basic trained network (24 *training sets* × 3 *initializations* × 2 *ANN architectures*) for ROC experiments on simulated data (provided the ANN is able to converge on a solution for all 144 trainings). Since we have only 15 training sets of real data, we decided to perform 5 random initializations of the ANN weights prior to training the network. Therefore, we may have up to 75 instances of a basic trained network (15 *training sets* × 5 *initializations*) for ROC experiments on real data. So, overall we will compare the two methods of ROC curve generation for up to 219 separate instances of a basic trained network.

2.2.4 The current standard ROC generation method: output node thresholding

The current accepted method to generate ROC points for ANNs is simply to vary the threshold on the value produced at the output node [20, 21, 22, 23]. The threshold (T_{out}) is varied over the range of the output node value (0.0 to 1.0). For each of the values of T_{out}, any feature vector which produces an output greater than or equal to T_{out} is labeled a target, otherwise it is labeled a non-target.

Recall that we must use the training data to estimate how ANN parameter values must be varied in order to produce a set of ROC points. In this case, we need to estimate a set of T_{out} values that correspond to various levels of sensitivity. This can done by simply sorting the set of output values that are found when every sample in the training set is input to the trained network. Each *distinct* value in this sorted set corresponds to a T_{out} value that will produce a new point on a ROC curve plotted *for the training set.* A set of T_{out} values determined in this manner produces the maximum number of ROC points that can be directly found. Additional points, if desired, would be found by interpolating between actual points found from the training data. For example, assume the training data gives (TP%,FP%) operating points of (40%, 10%) for an output threshold of 0.6, and (50%, 20%) for an output threshold of 0.7, but no points in between these. An output threshold of 0.65 might be assumed to produce an interpolated ROC point of (45%, 15%). Note that this newly estimated value of T_{out} will not produce a new ROC point for the training set, but it may well result in a new ROC point for a data set with a similar distribution as the training set data (such as the test data).

For the experiments reported here, we first find all of the distinct T_{out} values for the training set. Then, using the interpolation procedure described above, we estimate a set T_{out} values corresponding to TP rates ranging from 0% to 100% in 1% intervals[2]. The goodness of this set of 101 T_{out} values as a ROC curve is then evaluated using the test set. Of course, it would be possible to generate an "optimal" ROC curve for the test data by selecting T_{out} values from the sorted set of output values that are found when the test set is input to the trained network. However, this would introduce the possibility of bias since classifier parameters would be determined directly from test data. In our work, *all* ANN parameters, connection weights and thresholds, are learned from training data only. Test data is used strictly for performance evaluation.

2.2.5 The proposed ROC generation method: scaling the bias weights for the first hidden layer

Consider a basic trained network. At each node, the weighted sum of inputs to that node becomes the input to a sigmoid function which determines the output value for the node:

$$\textit{node output value} = \frac{1}{1 + e^{-(W_1X_1 + W_2X_2 + ... + W_dX_d + W_0B)}} \tag{7}$$

In this expression, the $X_i, i = 1...d$, are the inputs (other than the bias) to the node, the $W_i, i = 1...d$, are the weights on the inputs, and W_0 is the weight on the bias input. The bias input, B, is fixed at 1.0 during the backpropagation learning phase. Scaling the bias input weight to make it greater than the value learned during network training will result in a greater value for the output of the node for a given set of input values. Scaling the bias input weight to make it less than the value learned during training has the opposite effect, translating into a reduced value for the node output for a given set of input values. Depending on the overall configuration of weights learned for the network, scaling the bias

[2] In principle, any desired number of T_{out} values with any predicted spacing in TP rates can be created by appropriate interpolation between actual T_{out} values that are found directly from the training data. A set of 101 values with 1% intervals in TP rate was judged sufficient for our purposes here.

weight for a given node to make it greater could either increase or decrease the value produced at the output node for a given input feature vector.

The mechanics of the proposed method of generating ROC curves are as follows. The first step is to determine, separately for each first hidden layer node, whether it is necessary to increase or decrease its bias input weight in order to cause more samples to be classified as targets (i.e increase TP and FP rates). This is accomplished in the following manner. The training set data is applied to the basic trained network[3], and the resulting TP and FP rates are noted. This TP/FP pair represents the "natural" ROC point to which the ANN has trained. Now, for a single first hidden layer node, the bias input weight is increased by an arbitrary increment, the training set is applied, and the resulting TP and FP rates are observed. Next, the bias input weight is decreased by an arbitrary amount, the training set is applied, and the resulting TP and FP rates are observed. This process is repeated for each first hidden layer node.

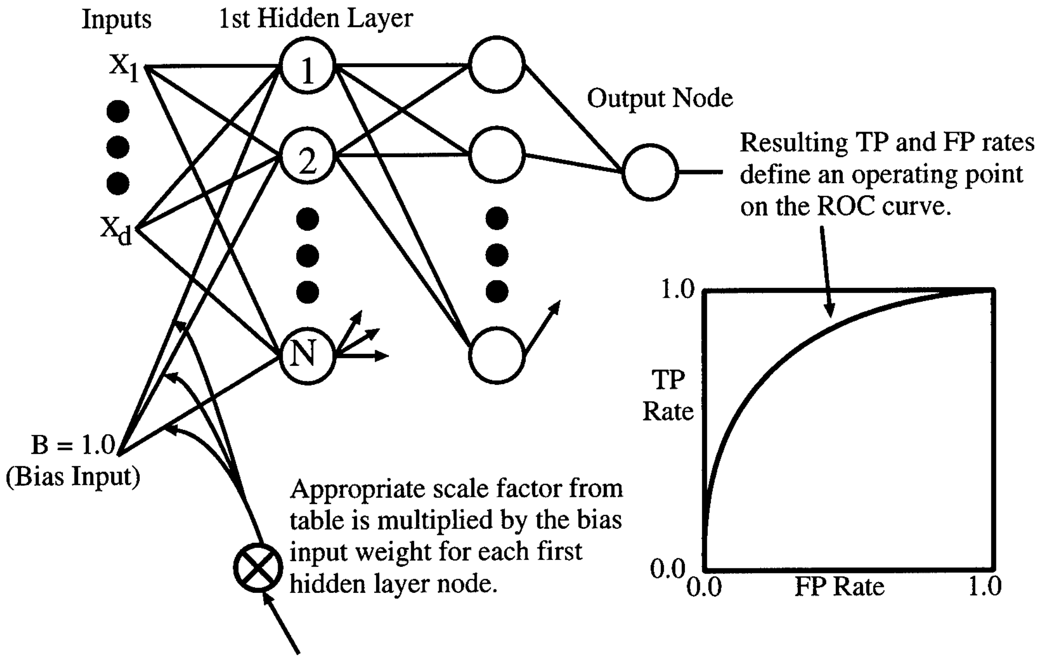
The bias input weight for a single node is increased or decreased by multiplying the weight (W_0) by a scale factor. In our implementation, this is done by setting the bias input to each node. During training the bias input is kept at a constant value of 1.0 for all nodes in the ANN. In our method of generating a ROC curve for the ANN, we vary this value individually for each node on the first hidden layer. In effect, the bias value at a given node becomes a "scale factor" for the bias weight learned at that node. We use scale factors of -9.0 and 11.0 to respectively decrease and increase bias input weights for this first step of the algorithm. We should note here that for some nodes the TP and FP rates are not affected when the bias input weight is changed. These nodes have effectively been "turned off" during training and are playing no role in determining the network output. Such nodes are not considered in ROC curve generation. So, at this point we know which "direction" (increasing, decreasing, or not at all) that each first hidden layer node must be scaled in order to increase or decrease both the TP and FP rates.

A set of scale factors for the bias weights on the first hidden layer will change the network operation and lead to a new (TP,FP) rate on the training set. In this view, the originally learned weights can be viewed as having implicit scale factors of 1.0. Since ANN training results in an initial specific point on the ROC curve, generating the rest of the curve involves changing the TP rate (sensitivity) of the ANN and observing the corresponding FP rate. Thus, the next step to generate a ROC curve is to determine sets of scale factors (using the training set) that change the ANN's TP and FP rates in a desirable manner. Our algorithm achieves this in two steps. First, we determine sets of scale factors that increase the TP rate from the initial ROC point while increasing the FP rate as little as possible (or not at all). Second, we determine sets of scale factors that decrease the TP rate from the initial ROC point while decreasing the FP rate as much as possible. Conceptually, we are "sweeping out" a ROC curve by attempting to find operating points for the training data that change the TP rate from 0% to 100% while maintaining as low a FP rate as possible.

We will now describe our implementation for determining sets of scale factors and applying them to generate a ROC curve. A lookup table data structure is created. The table has a row for each first hidden layer node. Each column in the table contains the set of scale

[3] Applying a data set to an ANN implies that each sample in the set is input to the network and classified according to its output value.

factors that correspond to a particular operating point found using the training set data. The lookup table, when completed, specifies how to generate a set of ROC points from the basic trained network (see Figure 3).


Inputs
1st Hidden Layer
X_1
X_d
1
2
N
Output Node
Resulting TP and FP rates define an operating point on the ROC curve.
B = 1.0
(Bias Input)
Appropriate scale factor from table is multiplied by the bias input weight for each first hidden layer node.
1.0
TP Rate
0.0
0.0
FP Rate
1.0

Hidden Node #	Scale Factors for 1st Hidden Layer Nodes. Operating Points as (TP rate, FP rate).				
	(0,0)	···	(TP,FP)	···	(100,100)
1		···	X.XX	···	
2		···	X.XX	···	
		···		···	
N		···	X.XX	···	

Select desired operating point from the lookup table.

Figure 3: The lookup table of scale factors is used to set a desired operating point for the ANN classifier. In this figure, the bias connections to the second hidden layer and output nodes are not shown. The bias value for these nodes is always 1.0.

The following algorithm is used to fill in the lookup table. The lookup table is initialized with the middle column entries all set to 1.0. These are the default scale factors for the bias input weights for the first hidden layer nodes in the basic trained network. Thus, this middle column represents the operating point to which the ANN naturally trained. Starting at the middle of the lookup table, the scale factors for all first hidden layer nodes are moved *simultaneously* by the same amount in the direction that will increase the TP and FP rates. Based on the first step in our algorithm we know which direction the scale factors should be changed for each first hidden layer node, but not by how much. We would like to be able to change the TP rate in as fine of increments as possible. So, we need to change the scale factors until at least one more target in the training set is correctly classified as a target (i.e. a new TP is found), thereby raising the TP rate. We perform a "binary search" over a range of possible scale factor changes until we zero in on the smallest scale factor change (within some desired degree of accuracy) that increases the TP rate for the training set. This process is depicted in Figure 4. The binary search technique is an efficient way to limit the number potential scale factor changes examined. This is important since the full set of training data must be applied to the network to get a TP and FP rate for each scale factor that is examined. Once the new set of scale factors have been determined, the new TP and FP rates of this operating point are recorded.

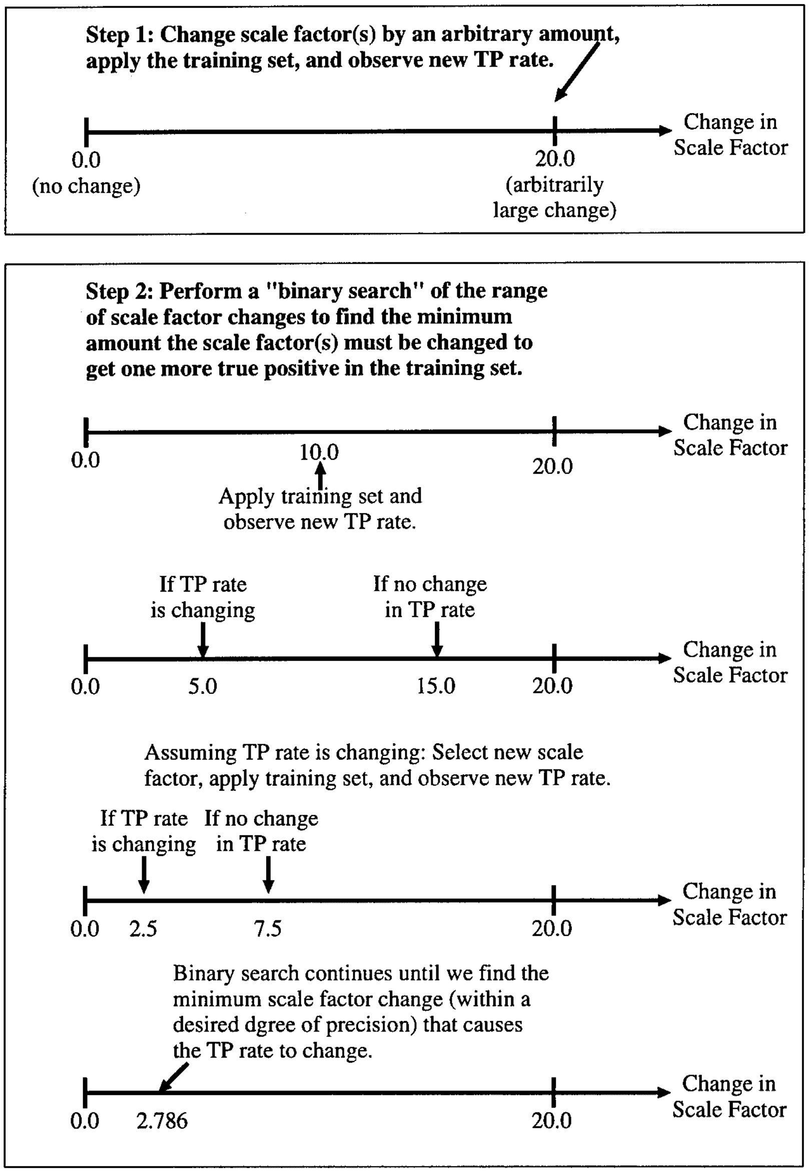

Step 1: Change scale factor(s) by an arbitrary amount, apply the training set, and observe new TP rate.
0.0
(no change)
20.0
(arbitrarily large change)
Change in Scale Factor
Step 2: Perform a "binary search" of the range of scale factor changes to find the minimum amount the scale factor(s) must be changed to get one more true positive in the training set.
0.0
10.0
20.0
Change in Scale Factor
Apply training set and observe new TP rate.
If TP rate is changing
If no change in TP rate
0.0
5.0
15.0
20.0
Change in Scale Factor
Assuming TP rate is changing: Select new scale factor, apply training set, and observe new TP rate.
If TP rate is changing
If no change in TP rate
0.0
2.5
7.5
20.0
Change in Scale Factor
Binary search continues until we find the minimum scale factor change (within a desired dgree of precision) that causes the TP rate to change.
0.0
2.786
20.0
Change in Scale Factor

Figure 4: A "binary search" is used to search a large range of potential scale factor changes.

If the density of the training set points in feature space is the same near all regions of the decision boundary, then changing the scale factors of all nodes simultaneously makes sense. However, if the density of training set points is different near different regions of the decision boundary, then it may be better to scale for only a selected subset of the bias weights. To check for this possibility, the scale factor is changed in the appropriate direction for each first hidden layer node *individually* (using the binary search techniques described above) until a minimal increase in the TP rate is found. Again, we record the new TP and FP rates.

So, if we have N nodes on the first hidden layer, then there are $N+1$ possible sets of scale factors, and therefore $N+1$ candidate operating points, from which to choose. A simple rule suffices to select the best operating point from among these candidates. Select the candidate operating point which increases the TP rate by the least amount (i.e. the finest increment in TP rate possible). If more than one candidate point produces this minimal change in the TP rate, then select the one of these that increases the FP rate by the least amount. When there are ties (i.e. more than one set of scale factors produces the best operating point), changing the scale factors for all first hidden layer nodes simultaneously takes precedence over changing the scale factor for a single node, and ties between changing different individual node scale factors are resolved arbitrarily. Finally, the next column in the lookup table is filled in with the scale factors that correspond to the selected ROC point. This procedure of increasing the TP and rate and filling in the lookup table is continued until we get a TP rate of 100% on the training set.

A similar strategy is employed to fill in the columns of the lookup table to the left of the middle column. Similar to before, we change scale factors for all first hidden layer nodes *simultaneously*, and for each first hidden layer node *individually* to generate a set of possible new operating points. From the $N+1$ candidate operating points, we select the one that reduces the TP rate the least. If more than one candidate point produces this minimal reduction in the TP rate, we select the one of these that reduces the FP rate the most. Ties are broken in the same manner as before. The scale factors for the newly selected operating point are added to the lookup table. We continue to decrease the TP rate and fill in the lookup table until either a TP rate or a FP rate of 0% is found for the training set.

At this point, we have a lookup table which gives the scale factors required for each operating point on the ROC curve based on the training set data. An example of a portion of an actual lookup table is shown in Table 1. Notice that sometimes all scale factors are changing simultaneously between successive operating points, and sometimes only a single scale factor is changing. Also, moving from left to right, notice that the scale factor increases for some nodes and decreases for others.

We should note that our solution for obtaining sets of scale factors for the bias weights is a heuristic. It chooses between manipulating either the scale factor for some one first hidden layer node or for all first hidden layer nodes at once. It is possible that better operating points may be found by changing more than 1 but less than all scale factors. Considering all possible subsets would require checking 2^N combinations, where N is the number of first hidden layer nodes. An exhaustive search would be out of the question for large N. The heuristic of choosing from among all hidden layer nodes or any single node should generally give good classification performance at reasonable computational cost. As a more sophisticated heuristic, a genetic algorithm could be used to help determine scale factor changes for varying numbers of first hidden layer nodes.

Table 1: Portion of a lookup table that specifies the scale factors for the bias weight inputs of each first hidden layer node. Each column in the lookup table corresponds to an operating point found for the training set data.

Scale Factors to Decrease < – – – – < – – – – < – – – – TP and FP Rates					Default Scale Factors	Scale Factors to Increase – – – – > – – – – > – – – – > TP and FP Rates			
Hidden Unit #	... ...	TP=65.3 FP=12.0	TP=67.0 FP=12.7	TP=68.0 FP=14.0	TP=68.7 FP=16.0	TP=71.3 FP=17.3	TP=72.7 FP=21.3	TP=76.0 FP=28.0	... ...
Unit 9	...	1.0	1.0	1.0	1.0	1.0	1.06	1.06	...
Unit 10	...	1.0	1.0	1.0	1.0	1.1	1.16	1.26	...
Unit 11	...	0.97	0.97	0.97	1.0	1.0	1.06	1.06	...
Unit 12	...	1.6	1.6	1.0	1.0	1.0	0.94	0.94	...
Unit 13	...	1.0	1.0	1.0	1.0	1.0	1.06	1.06	...
Unit 14	...	1.0	1.0	1.0	1.0	1.0	1.06	1.06	...
Unit 15	...	1.0	1.0	1.0	1.0	1.0	1.06	1.06	...
Unit 16	...	1.0	1.0	1.0	1.0	1.0	1.06	1.06	...
Unit 17	...	1.0	1.0	1.0	1.0	1.0	1.06	1.06	...
Unit 18	...	0.94	1.0	1.0	1.0	1.0	1.03	1.03	...

From the sets of scale factors determined for the training set, we can interpolate additional sets of scale factors for projected ROC points for a new set of data. Once the lookup table has been completed for the training set, we interpolate between sets of scale factors for successive ROC points to find a set of scale factors that could theoretically generate 101 ROC points (0% to 100% TP rate in 1% increments). For example, using Table 1 and interpolating to get an operating point with a predicted TP rate of 75%, the scale factor for node 10 would be approximately 1.23, while the scale factors for the other nodes remain unchanged. These 101 sets of scale factors are used to generate the ROC curve for the test set.

2.2.6 Comparing Methods of ROC Generation

Both the standard method and our proposed method begin with the basic trained ANN. For each method, the training set is used to generate a predicted set of 101 ROC points at 1% increments in the TP rate. The performance of the methods is then evaluated on the separate test set.

For our experimental results, we will compare AUCs in three ways. First, we make an absolute (no statistical test) comparison of AUCs to determine which method generates a better ROC curve most of the time given a trained ANN. Testing our statistical hypothesis using a two-tailed test helps to determine if the difference between two observed AUCs is significant. Since the ROC curves are derived from the same data sets, we should consider a correlation factor when computing the test for statistical significance [24]. If we do not consider a correlation factor, the difference in AUCs must be relatively large in order to pass the test. A non-zero correlation factor increases the chances of finding a statistically significant difference in AUCs. Our second way of comparing AUCs is a very stringent test in which we do not consider a correlation factor. Finally, our third way of comparing AUCs, which is probably the most accurate method of comparison, considers the correlation factor in the calculations of the critical value z.

In addition to the AUC measurement, we also compare the number and range of distinct operating points that each ROC method generates on the test data. A greater number of distinct operating points means there is more flexibility when selecting a desired TP and FP rate. A finely sampled range of operating points is important because a different TP/FP trade-off may be needed for different applications. A: ROC curve with a "full range of operating points" has relatively small differences in *either* the TP rate or the FP rate between any two consecutive points on the curve. As evaluated here, both methods of ROC curve generation have the same potential to produce 101 evenly spaced ROC points. That is, parameters are generated based on the training set to give 101 distinct projected operating points. These 101 parameter settings may or may not generate distinct points when used in an evaluation with the test set.

2.2.7 Results

For the 219 training attempts described in Subsection 2.2.3 (144 for simulated data and 75 for mammography data), the ANN failed to converge 41 times. Thus, we have 178 instances of a basic trained network for the purpose of comparing the ROC curve generating capabilities of the two methods. All of the experimental results are summarized in Table 2.

Simulated Data:

For the simulated data, the ANN failed to converge on a reasonable solution 24 times, or about 17% of the time. Our proposed method generates a greater AUC for nearly 87% (104 of 120) of the trained network instances. The conventional method generates a greater AUC about 12% of the time. The two methods generated equivalent AUCs less than 2% of the time. Our method has a statistically significantly greater AUC for about 81% (97 of 120) of the individual instances if the correlation factor is considered. The AUCs are not statistically significantly different (i.e. statistically equivalent) for 19 instances, or about 16% of the time. Of the 120 instances, the conventional method generates a significantly better AUC only 4 times, or about 3% of the time. Even if the correlation factor is not taken into account, Table 2 shows that our method is still considerably better.

The average number of distinct operating points that were generated from the 101 ROC points projected from the training data are 12 and 44 for the current standard method and our proposed method, respectively. In general, fewer operating points were generated for the simulated data than for the mammography data. Since the simulated data sets have more "ideal" distributions than the mammography data, the ANNs were able to find relatively good solutions during training. In fact, the AUC values are typically greater than 0.96 for the simulated data, whereas the AUC values for the mammography data are generally around 0.90. As a result, many of the ROC curves generated for the simulated data usually have an operating point with a high TP rate (> 80%) and a 0% FP rate, and/or a point with a 100% TP rate and a FP rate well below 100% (around 40%). Thus, no ROC points for a

Table 2: Summary of test results indicating the average number of ROC points generated by each method, and the number of instances each method produces a better ROC curve. M1 is the conventional method of generating ROC curves by thresholding the value at the output node. M2 is our proposed method that scales the bias input weights of the first hidden layer nodes.

Data Type	# of ANNs That Did Not Train	# Basic Trained Networks	Average # of ROC Points		Direct Comparison of AUCs		
			M1	M2	M1 Better	M2 Better	Equivalent AUCs
Simulated	24	120	12	44	14	104	2
Mammography	17	58	52	86	11	47	0
Totals	41	178	17	58	25	151	2

Data Type	# of ANNs That Did Not Train	# Basic Trained Networks	Average # of ROC Points		Significance Test Without Correlation		
			M1	M2	M1 Better	M2 Better	Equivalent AUCs
Simulated	24	120	12	44	4	91	25
Mammography	17	58	52	86	1	13	44
Totals	41	178	17	58	5	104	69

Data Type	# of ANNs That Did Not Train	# Basic Trained Networks	Average # of ROC Points		Significance Test With Correlation		
			M1	M2	M1 Better	M2 Better	Equivalent AUCs
Simulated	24	120	12	44	4	97	19
Mammography	17	58	52	86	5	31	22
Totals	41	178	17	58	9	128	41

large range of TP or FP rates are needed. For example, if we get a TP rate of 80% with an FP rate of 0%, then there is no need to generate operating points with TP rates less than 80%. Similarly, if we get a TP rate of 100% with a FP rate of 40%, there is no need for operating points with FP values greater than 40%.

Mammography Data

The ANN failed to converge on a reasonable solution 17 times, or about 23% of the time, for the mammography data. This is a slightly higher rate than for the simulated data. This is not surprising if we consider the simulated data is more "ideal" than the mammography data. Our proposed method generates a greater AUC for about 81% (47 of 58) of the trained network instances. The conventional method generates a greater AUC about 19% of the time. The two methods did not generate an equivalent AUC for any of the 58 instances. Our method has a statistically significantly greater AUC for about 53% (31 of 58) of the individual instances if the correlation factor is considered. The AUCs are not statistically significantly different for 22 instances, or about 38% of the time. Of the 58 instances, the conventional method generates a significantly better AUC only 5 times, or less than 9% of the time. As before, if the correlation factor is not considered, our proposed method still outperforms the conventional method. The average number of operating points found for the current standard method and our proposed method are 52 and 86, respectively.

Distribution of ROC points

The current standard method generally had problems in finding a full range of operating points. As an example, for one of the basic trained ANNs an operating point, denoted by (TP rate, FP rate), was (9.6%, 0.52%), but the next closest operating point was (85.4%, 22.4%). So, operating points could not be obtained in this case for almost 76% of the possible TP operating range or about 22% of the possible FP operating range. Therefore, it may not be possible for an ANN to operate near a desired TP or FP rate if the current standard method is used to generate operating points. This phenomena, illustrated in Figure 5, is characteristic to some degree of many of the ROC curves generated using the current standard method, and it is a serious drawback. Our proposed method is always capable of producing a full range of operating points.

2.3 CMC Using Local Accuracy Estimates

Section 2.3.1 introduces our CMC algorithm and describes some potential variations. Section 2.3.2 briefly describes the other CMC algorithms which we have implemented and tested. Section 2.3.3 describes the experimental procedures, covering the data, the individual classifiers, and some implementation issues. Section 2.3.4 compares the performance of the individual classifiers and the CMC algorithms.

2.3.1 The DCS-LA Approach

Dynamic Classifier Selection (DCS) attempts to determine which classifier is most likely to make a correct decision for a given test sample. One straight-forward approach is to estimate each classifier's accuracy in a region of feature space surrounding the test sample. We term our approach to CMC as Dynamic Classifier Selection by Local Accuracy, or DCS-LA. Local accuracy estimates (LAEs) can help determine where in feature space a particular classifier performs most reliably.

There are two important issues that must be addressed in order to implement a DCS-LA algorithm. First, how should the local region about a test sample be defined? Second, how should the local accuracy be estimated?

A local region about a sample in feature space may be defined in any of several ways. One approach is to take the K-nearest neighbor training samples to the test sample. This approach defines a local region as the convex hull of the K nearest training samples. It is not necessarily clear, a priori, what region size (value of K) would be best for a particular problem.

Once a local region has been defined, the accuracy of a classifier within the region can be estimated from either the training data or a separate set of validation data. One possible estimate of local accuracy would be simply the percentage of training samples in the region

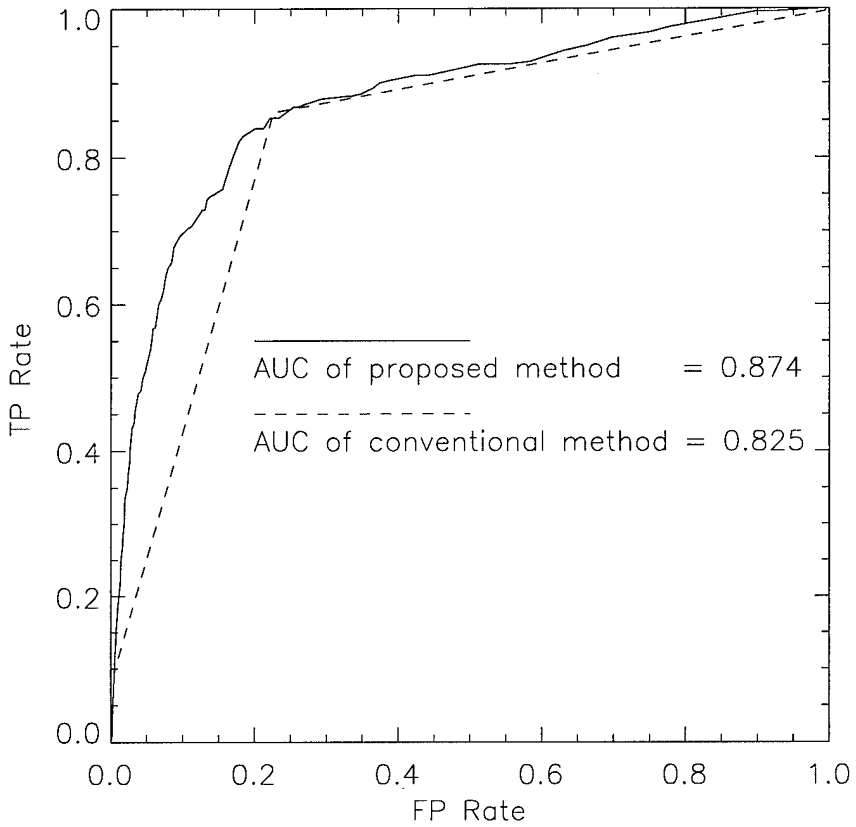


Figure 5: ROC curves generated by both methods for the same training and test sets of mammography data. The difference between the AUCs of the ROC curves generated by both methods is not always this large. This particular example was selected to illustrate a phenomena that is characteristic of the conventional method. Here, the conventional method was unable to generate any operating points with FP rates between 0.52% and 22.4%.

that are correctly classified. We'll refer to this as the overall local accuracy. It is an estimate of the probability that the classifier is correct in some small pocket of feature space.

Another possibility would be to estimate local accuracy with respect to some output class. Consider a single classifier that assigns a test sample to class C_i. We can determine the percentage of the local training samples assigned to class C_i by this classifier that have been correctly labeled. We'll refer to this as the local class accuracy. It more specifically estimates the probability that the classifier is correct in the small pocket of feature space when making a particular decision.

Our DCS-LA implementation defines the local regions as the K-nearest neighbors in the training sets. Both methods described above for computing LAEs will be examined.

2.3.2 Algorithms for Comparison

In order to determine the relative merit of our CMC algorithm, we have selected two previously published algorithms [8, 12] for direct comparison.

The Behavior-Knowledge Space Approach

The Behavior-Knowledge Space (BKS) algorithm has recently been tested on an application for recognizing unconstrained handwritten numerals. Huang and Suen [8] show the BKS method to be superior to voting, Bayesian, and Dempster-Shafer approaches.

Behavior-Knowledge Space is a N-dimensional space where each dimension corresponds to the decision of one classifier. Each classifier can assign a sample to one of M possible classes. Depending on the application, a classifier may have $M+1$ possible decisions, where class $M+1$ is a "reject" decision. In our application, we do not permit rejections, and so each classifier has M possible decisions. Each unit of a BKS represents a particular intersection of individual classifier decisions. Thus, the BKS represents all possible combinations of the individual classifier decisions. Each BKS unit accumulates the number of training samples from each class. For an unknown test sample, the decisions of the individual classifiers index a unit of BKS, and the unknown sample is assigned to the class with the most training samples in that BKS unit[4]. The BKS method is easy to implement, easy to train, and computationally efficient in the testing mode.

Classifier Rank Using a KNN Approach

In [12], Sabourin et al. present a DCS algorithm which proved to have some similarities to our DCS-LA approach. From the training data, they extract a set of "correctness" features: the Euclidean distance to the closest misclassified sample, the Euclidean distance to the closest correctly classified sample, and the ratio of these two distances. They note which classifier(s) correctly classified each training sample. For an unknown test sample, the three correctness features are computed, and the nearest neighbor (NN) training sample in this "new" feature space is found. The classifier that correctly labeled the NN training sample is dynamically selected to classify the unknown sample using the original feature data.

A better performing variation of their algorithm selects the classifier that correctly classifies the most consecutive neighboring training samples (relative to the unknown test sample). The selected classifier is said to have the highest "rank". Although they do not associate their DCS algorithm with the concept of local accuracy, their notion of classifier rank certainly has the flavor of a LAE. We will refer to this algorithm as the Classifier Rank method, or CR.

An Alternate LAE

In terms of our work, the CR algorithm presented in [12] uses a LAE which we would describe as an overall local accuracy estimate. An obvious alternative would be to use local class accuracy for the LAE. Thus, we implemented a version of our DCS-LA algorithm which incorporates the ideas of classifier rank and local class accuracy as a LAE. Given a

[4] In the event that a tie exists in a BKS unit, our implementation dynamically selects the output of the most globally accurate individual classifier. As it turns out, this tie-break was very rarely needed, and the overall effect was negligible.

test sample assigned to class C_i by a classifier, the LAE for the classifier is computed as the number of consecutive nearest neighbors assigned class C_i which have been correctly labeled. We will refer to this variation of our algorithm as DCS-LA/2.

2.3.3 Experimental Procedures

For a CMC approach to be of practical use, it should improve on the best individual classifier, given that the individual classifiers have been reasonably optimized. This is necessary in order to ensure that improved performance for the CMC algorithm is in fact due to the combination of the classifiers rather than to incomplete training or design of the individual classifiers. Few published CMC works are clear as to how well the individual classifiers have been optimized. (We have found one paper [5] that shows a clear attempt to construct individual classifiers with all the information available to the CMC algorithm.) In our work, each classifier has the potential to draw from the same large set of features. Also, effort is made to optimize each individual classifier with respect to selecting "good" values for the parameters which govern its performance.

Individual classifiers can be set for various true positive and false positive rates[5]. Performance of a CMC algorithm as a function of the true positive or false positive rates of the individual classifiers has not previously been examined.

Data Sets

The task is to detect abnormalities in mammograms, labeling pixels as either "normal" or "abnormal" tissue. Thus, we have a 2 class problem. The DCS-LA algorithm is also applicable to multi-class problems such as character recognition. Additionally, multi-class problems can be defined in terms of 2 classes by making a binary (yes or no) decision for every class.

A data set of 40 digitized mammograms[6] containing some abnormalities was divided into two sets of images, Set A and Set B. From each set of images, pixels from the abnormal and normal class were randomly sampled. Set A has 19,735 samples from the normal class, and 3001 samples from the abnormal class. Set B is made up of 20,028 normal samples and 5159 abnormal samples. For each pixel, 63 features were computed. A more detailed description of this feature data can be found in [25]. The images from which the feature data was extracted were provided by Nico Karssemeijer, and are used in his previous research [26]

Initially, Set A is used as training data for the individual classifiers and the CMC algorithm, and Set B is used to measure performance. Then the roles of Sets A and B are reversed. Thus, at no time are samples from the same image used for training *and* testing in the same set of experiments. Whenever we talk of training (or test) data, both Set A and Set B have been utilized independently in that capacity. Thus, the results of feature selection, individual classifier performance, and CMC results should be expected to be similar, but not identical, for Sets A and B.

[5] This terminology is often associated with 2-class problems. Other multi-class problems, like handwritten character recognition, use the analogous terms recognition and substitution rates.

[6] Images were provided by courtesy of the National Expert and Training Centre for Breast Cancer Screening and the Department of Radiology at the University of Nijmegen, the Netherlands.

Individual Classifiers

We use six individual classifiers as input to the various CMC algorithms, two parametric and four non-parametric. They are Linear Bayesian (LC), Quadratic Bayesian (QC), K-Nearest Neighbor (KNN), two decision tree implementations (BDT and C4.5), and an artificial neural network (ANN). Feature selection has been performed for each of the six classifiers. The details and results of the feature selection process have been omitted from this report.

DCS-LA Implementation and Application

For each classifier, the DCS-LA algorithm uses the training data, and the final decision made by the classifier for each training sample. The individual sample inputs, which may be different for each classifier, are needed to find the neighboring samples to an arbitrary test sample in each classifier's feature space. Obviously, the class assignments made by each classifier are needed to determine local classifier accuracy.

Once all the training data has been loaded, we are prepared to classify an unknown sample. First, the sample is labeled by all the individual classifiers. If all classifiers agree, there is no need to compute LAEs. When the individual classifier disagree, a LAE is computed for each classifier, and we select the decision of the classifier with the highest LAE.

Occasionally, two (or more) classifiers with conflicting decisions will have the highest LAE. Tie breaking is handled by choosing the class that is selected most often among the tied classifiers. If a tie still exists, the classifier(s) with the next highest LAE(s) will break the tie in the same manner as before.

Since determining the appropriate size for a "local" region is part of designing the DCS-LA approach, we need to test a range of local region sizes. We ran experiments for 10 different region sizes: K = 1, 5, 10, 15, 20, 25, 30, 25, 40, and 50. Here, we utilize the Mahalanobis distance metric [27] to find the K-nearest neighbors, as this adapts to features that are measured on different scales.

We would also like to investigate the effect of setting the individual classifiers to various TP rates prior to applying CMC. We tested all CMC algorithms with the individual classifiers set to 6 different TP rates: 70%, 75%, 80%, 85%, 90%, and 95%. Not all of the classifiers could be set to the exact TP rate desired. In these cases, the individual classifiers were set as close to the desired TP rate as possible. Overall we generated 60 operating points for each of the two DCS-LA variations: 10 region sizes with individual classifiers set to 6 different TP rates.

2.3.4 Classification Results

We now present test results for the individual classifiers and the various CMC algorithms. For brevity, the results obtained when Set A is used to train and Set B is used to test will be referred to as the E1 data (experiment 1). Similarly, the results obtained when Set B is used to train, and Set A is used to test will be referred to as the E2 data. Results for the E1 and E2 data are mostly similar. We will show experimental results for the E1 data, and note any differences for the E2 results.

Individual Classifiers

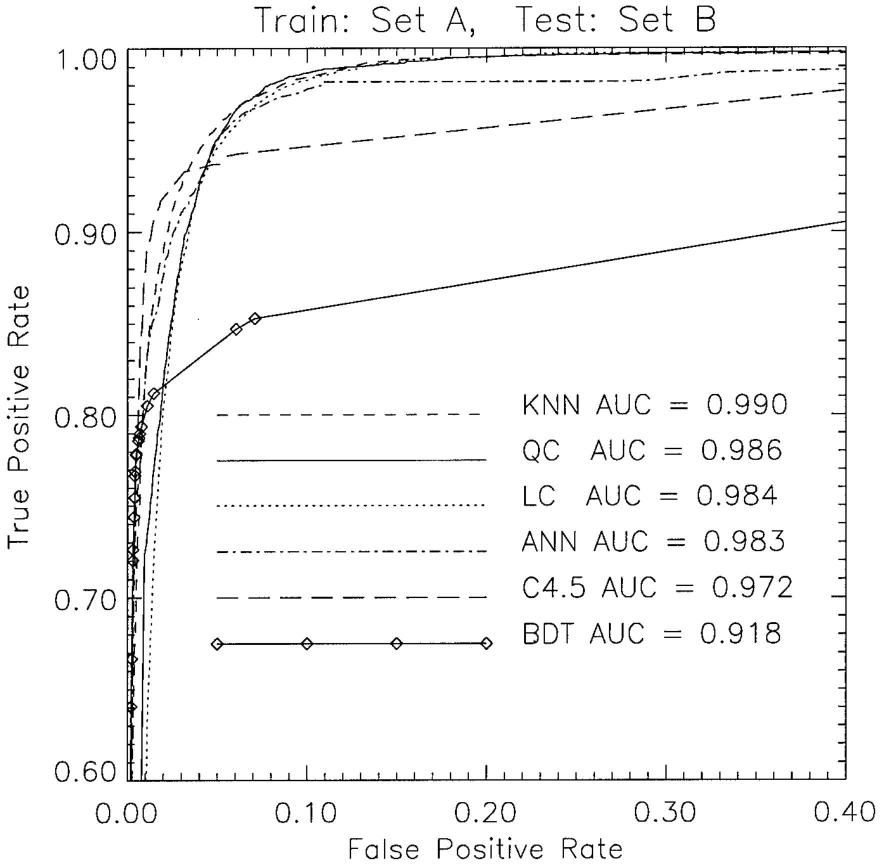


Figure 6: Partial ROC curves for the 6 individual classifiers, and their associated AUCs.

Figure 6 shows partial ROC curves[7] plotted for all 6 individual classifiers for the E1 data. Similar results are obtained for the E2 data. In both cases, the best individual classifier is KNN if the overall AUC is considered. However, there is no single best classifier for all TP rates. Consider the data in Figure 6. Depending on the desired TP rate, the QC, KNN, C4.5, or BDT could be considered the best individual classifier. From very low TP rates up to about a 79% TP rate, the BDT classifier has better FP rates than the other classifiers. For TP rates from 79% to about 92%, the C4.5 classifier is best. The KNN classifier has the lowest FP rates for TP rates from 92% to 100%, except in the small range from about 97% to 99%, where the QC classifier is superior. For the E2 data, depending on the desired TP rate, the best individual classifier is one of KNN, C4.5, or BDT.

As a benchmark for useful CMC performance, we consider a composite ROC curve consisting of the "best" parts of the individual ROC curves. This curve is constructed by

[7] Partial ROC curves are plotted as opposed to ROC plots that show the entire operating range (TP rates from 0.0 to 1.0 and FP rates from 0.0 to 1.0) in order to focus on a region of interest. In a medical application such as ours, high sensitivity levels are required.

considering the set of all the individual classifier operating points, and deleting inferior operating points. An operating point is deleted if and only if another point exists which has a lower FP rate *and* a higher TP rate. The composite ROC is a lower bound for practical CMC performance. We also plot ROC curves for an "oracle" classifier, which chooses the correct class if *any* of the classifiers did so. Thus, the only time the oracle cannot make a correct classification is when all the individual classifiers are wrong. Thus, the performance of the oracle is a theoretical upper bound for all CMC algorithms discussed in this work. The composite and oracle ROC curves for the E1 data are shown in Figure 7.

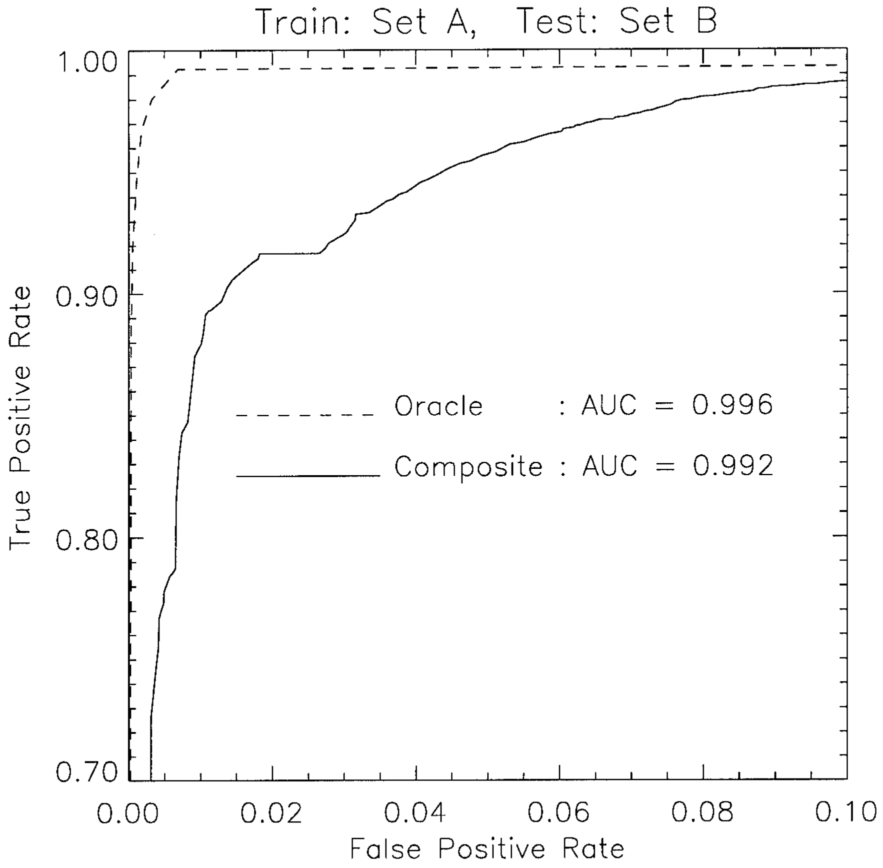


Figure 7: Composite ROC curve for the 6 individual classifiers, and the ROC curve for an oracle classifier.

CMC Algorithms

First, we would like to determine if it is better to compute the LAE as the overall local accuracy, or the local class accuracy. Recall, we generated 60 ROC points for both DCS-LA variations. In order to determine potential performance, we plotted partial ROC curves that use the best operating points available. As we discussed before, an operating point is ignored if and only if another point exists which has a lower FP rate *and* a higher TP rate. Clearly, in practice the operating point that is eliminated would never be used. Once the TP rates of the individual classifiers have been set, the DCS-LA algorithm will examine local regions of various sizes. Each region size will result in a slightly different operating point, and we may select whichever point or points are desired. Shortly, we will examine the algorithm's performance if we require all regions to remain a constant size regardless of the individual classifier settings.

The partial ROC curves for both DCS-LA variations using the E1 data are shown in Figure 8. From this figure, we can see that using local class accuracy as the LAE is superior to using the overall local accuracy. The difference is not statistically significant ($z = 1.44$ for TP rates ranging from 78% to 94%). Results for the E2 data confirm this.

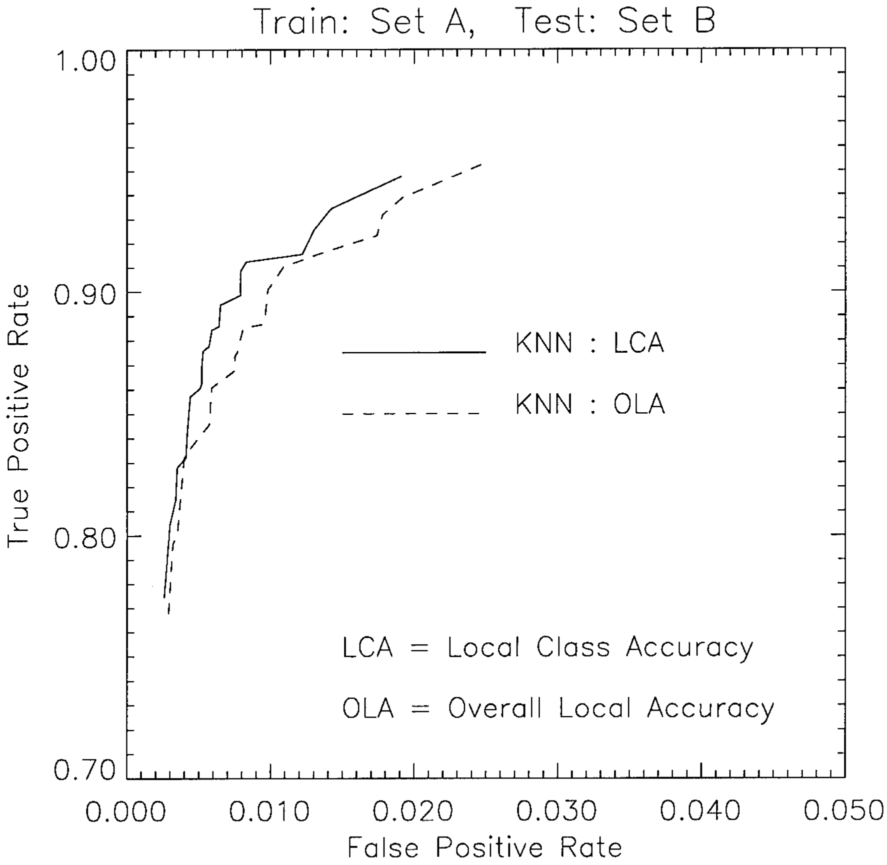


Figure 8: Partial ROC curves for the two DCS-LA variations. KNN refers to a local region defined as the K-Nearest Neighbors. The type LAE is denoted by either OLA or LCA.

Provided there is no significant difference in performance, a fixed value of K is conceptually simpler than testing for various values of K. Thus, we would like to examine whether smaller or larger regions are generally best. Figure 9 shows partial ROC curves for various size local regions. All plots are for the E1 data with the LAE computed as local class accuracy. This plot does not show results for every different region size, just enough to clearly permit some interesting observations. No single region size is obviously superior in all cases, although a region size using $K = 10$ is best a majority of the time. In general, region sizes with $K = 5$ or $K = 10$ seem to result in the best performance.

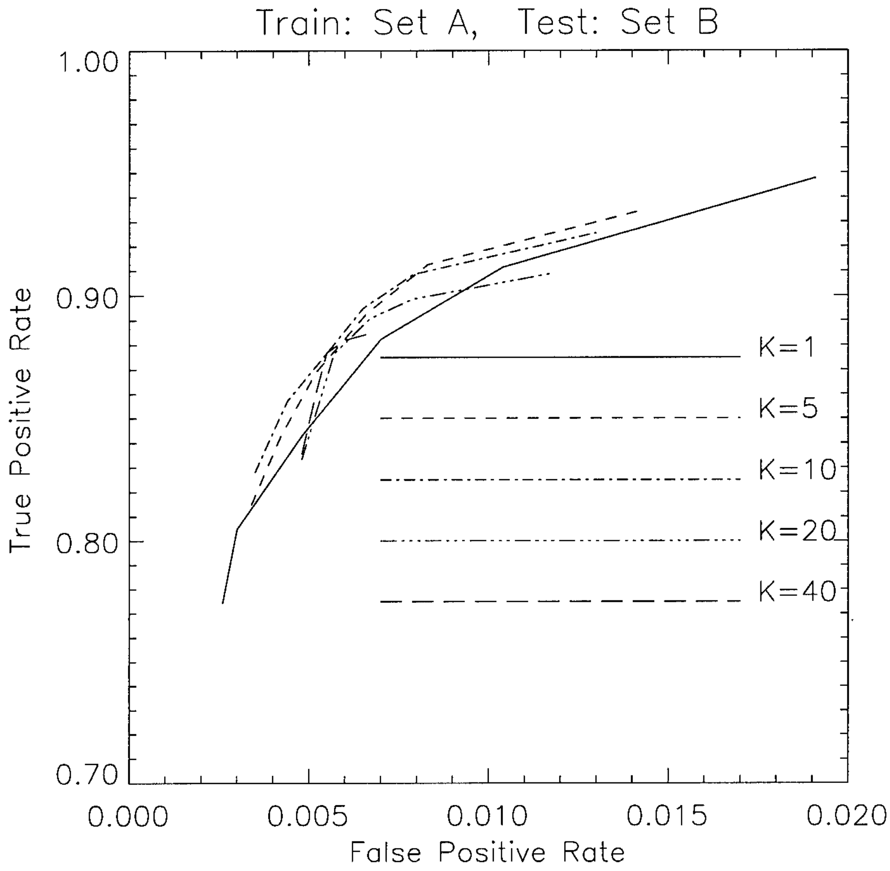


Figure 9: Partial ROC curves for DCS-LA algorithm with local regions of various sizes.

Now, we compare the performance of the best DCS-LA algorithm variation with that of the individual classifiers. Figure 10 shows the composite ROC curve (for the E1 data) for the 6 individual classifiers compared to the results for DCS-LA with the LAE computed as local class accuracy. We also show the results of the BKS, CR, and DCS-LA/2 algorithms. To be fair, only the best single value of K (10) is used in the plot for the DCS-LA results. Thus, the ROC curves for all four CMC algorithms are composed of 6 operating points each.

It is evident that our DCS-LA algorithm is better than the best individual classifier at all times. The difference between the AUCs, computed over the range of common TP points (from 82% to 93%), for DCS-LA ROC curve and the Composite ROC curve is statistically significant ($z = 3.51$). The DCS-LA/2 method performs nearly as well as DCS-LA at lower sensitivities, but less so at higher levels. It is significantly better than the best individual classifier ($z = 2.71$) over the common TP range (82% to 88%). The CR method provides improvement, though not statistically significant, at some levels of sensitivity. The BKS method is not able to improve upon the performance of the optimized individual classifiers. The DCS-LA method is the only CMC algorithm we tested that performed consistently better than the individual classifiers. It performed significantly better than the BKS method

(z = 4.91 for TP rates ranging from 84% to 92%), and the CR method (z = 3.81 for TP rates ranging from 82% to 91%).

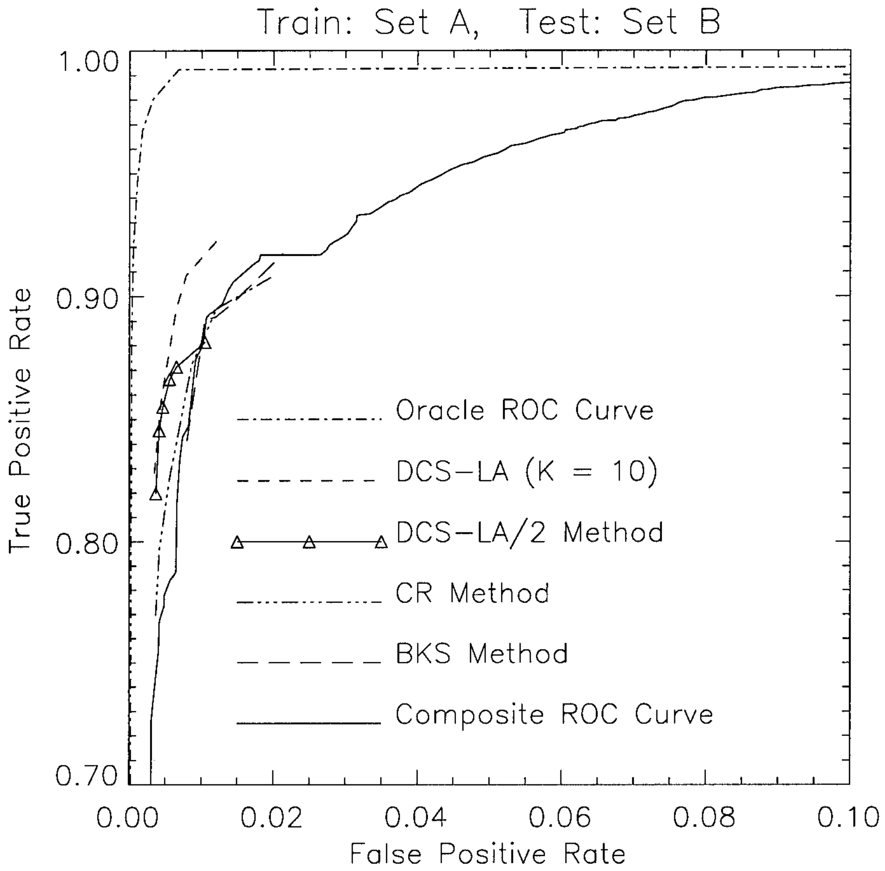


Figure 10: The composite and oracle ROC curves for the 6 individual classifiers compared to the results for the DCS-LA, BKS, CR, and DCS-LA2 methods.

With respect to our mammography application, for TP rates above 80%, the DCS-LA algorithm results in FP rates anywhere from 0.5% to 2% lower than the best individual classifier. To put this in perspective, consider that a single mammogram image may have on the order of 2 to 3 million pixels that require classification. So, for the same level of sensitivity (TP rate), we would misclassify anywhere from 10,000 to 60,000 fewer normal pixels than we would with the best individual classifier.

2.4 Contrast-Based vs. Texture-Based Segmentation

The most common approaches to segmenting potential lesions from mammogram images involves locating radiographically bright regions. Thus, the contrast of a lesion relative to surrounding background tissue is the feature upon which the segmentation is based. Alternatively, a segmentation routine based on the texture of a region can be used to extract potential lesions. The results shown in these subsections are for each segmentation routine alone. That is, there is no attempt to perform classification of image regions *after* they have been segmented. The following experiments have been run to determine which approach, if any, might offer some fundamental advantages in a mammogram image analysis application. All results are reported for the same set of 320 images, which contain a total of 62 lesions.

2.4.1 Contrast-Based Segmentation

Our first experimental results are for a contrast-based segmentation with region growing. This approach is described in detail in [17]. This approach represents an attempt to accurately segment a majority of all the lesions. That is, we have a very liberal segmentation routine in order to "fully" segment most of the lesions in the data set. Segmentation is based on local contrast in which pixels greater than the mean plus one-half a standard deviation of pixel intensities in a $10mm$ by $10mm$ square window are retained. There are some smoothing and median filtering pre- and post-processing operations. A rather complex region growing routine is used after the initial segmentation to improve shape estimation of the segmented object.
Results:
Average TP area per image : 32.1 %
Average FP area per image : 9.6 %
TP lesion detection rate : 100 % (62 of 62)
Total Number of FPs (320 images): 11172
Average Number of FPs per image : 34.9

2.4.2 Texture-Based Segmentation with One Feature

In the following three texture-based segmentation, an image is thresholded on one or more texture features, where the feature and the threshold have been selected empirically. We are not concerned with trying to fully segment each lesion. Thus, the is no region growing step after the thresholding operation. Additionally, the thresholds are global and absolute, so we may not segment many "pockets" of locally high or low feature values (which is virtually guaranteed to happen if we keep pixels according to mean and st.dev. of a local area, as before). Also, a pretty good size smoothing window ($3.5mm$ by $3.5mm$) is run after the thresholding operation.

The second segmentation routine is based on a single texture feature, relative extrema density. This feature many times responds with low values for a "ring" surrounding the lesion. A threshold on the texture feature was selected empirically such that we keep pixels with a feature value less than 37.0. As a smoothing step, a 3.5 mm square window is centered on each pixel. If more than 25thresholding step, then the pixel at the center of the window is kept.
Results:
Average TP area per image : 60.2 %
Average FP area per image : 12.0 %
TP lesion detection rate : 98.4 % (61 of 62)
Total Number of FPs (320 images): 2052

Average Number of FPs per image : 6.4

Basing the segmentation on a texture feature, as opposed to a contrast derived feature, appears to offer a tremendous improvement. We miss one lesion during segmentation, but the number of FP regions segmented is dramatically reduced.

2.4.3 Texture-Based Segmentation with Two Features

In this third routine, a pixel must pass thresholds for two texture features in order to be retained. In addition to thresholding pixels on the relative extrema density, as above, another texture feature called ALOE is computed for each pixel. Pixels with an ALOE feature value less than 0.0047 *and* a relative extrema density feature value less than 37.0 are retained. As before, the smoothing step says to keep the pixel at the center of a 3.5 mm square window if more than 25window survived both threshold operations.
Results:
Average TP area per image : 55.4 %
Average FP area per image : 9.8 %
TP lesion detection rate : 98.4 % (61 of 62)
Total Number of FPs (320 images): 1716
Average Number of FPs per image : 5.40

Adding a second feature to the texture-based segmentation scheme drops the FP rate even further, without reducing the sensitivity.

2.4.4 Texture-Based Segmentation with Three Features

In this routine, a pixel must pass thresholds for three texture features in order to be retained. The additional feature computed for each pixel is the average gradient (using the Sobel operator) in a 5mm by 5mm square window. Pixels with an ALOE feature value less than 0.0047 *and* a relative extrema density feature value less than 37.0 *and* an average gradient feature value greater than 225.0 are retained. As before, the smoothing step says to keep the pixel at the center of a 3.5 mm square window if more than 25inside the window survived all three thresholds.
Results:
Average TP area per image : 50.3 %
Average FP area per image : 6.12 %
TP lesion detection rate : 96.8 % (60 of 62)
Total Number of FPs (320 images): 1331
Average Number of FPs per image : 4.16

Adding yet another feature to the segmentation scheme drops the FP rate even further. This time there is a slight drop in sensitivity, as we fail to segment one less lesion.

3 Conclusions

3.1 ROC Analysis

We have completed a review of techniques for generating ROC curves for several families of statistical classifiers, and techniques for comparing two ROC curves. As a result, we noticed some fundamental weaknesses with the current approaches of generating ROC curves for non-traditional classifiers such as ANNs and decision trees. Additionally, we found no clearly defined techniques (which would apply to our work) for determining if the difference between portions of two ROC curves is statistically significant. By refining some previous work, we have developed a systematic method for comparing classification and image analysis algorithms. Thus, this research plays a key role in selecting individual components which will be incorporated into the final computer system.

3.2 Generating ROC Curves for ANNs

Two methods of ANN ROC generation are compared. These methods are (1) varying a threshold on the output node, and (2) scaling the bias input weight for selected first hidden layer nodes. Varying a threshold on the output node is the current standard method. Scaling the bias input weight for selected first hidden layer nodes is the new method developed through our research. We have shown that this new method produces statistically equivalent or significantly greater AUCs than those obtained with the current standard method over 90% of the time, and will generally result in a greater number of ROC points. Our proposed method involves the construction of a lookup table which contains a sequence of scale factors for the bias input weights of each first hidden layer node. The lookup table is used as a "sensitivity dial" which facilitates the easy selection of an operating point for an ANN classifier.

The main contribution of this work is that we have provided a method which allows ANNs to be used for reliable classification at operating points other than the single operating point for which they are trained. ANNs are generally trained to minimize the number of misclassifications or some error rate criteria. For applications where different types of errors have different costs, this "optimal" (in terms of error rate) operating point is not suitable. This additional flexibility is very much in the spirit of probabilistic classifiers which permit the selection of an operating point which can maximize "profits" when a "gain/loss" is associated with making a decision.

3.3 CMC Using Local Accuracy Estimates

We have developed a new algorithm for combining multiple classifiers that uses estimates of each individual classifier's local accuracy about a test sample. To classify an unknown sample, we determine which classifier is most accurate for a subset of training samples similar to the unknown sample. The output of the most locally accurate classifier is then used for classification. We have attempted to address some issues relevant to the construction of a multiple classifier system which have not previously received attention. These issues concern the optimization of individual classifiers, and the effect of varying the sensitivity of the individual classifiers on the CMC algorithm.

In all of our experiments, LAEs based on local class accuracy were more effective than those based on overall local accuracy. This LAE information can be used effectively as the selection mechanism for a CMC algorithm which dynamically selects the output of a single classifier to label a given test sample.

In our work, we made efforts to optimize the individual classifiers with respect to the available feature data. Certainly it would be preferable to use a single classifier as opposed to a combination of several classifiers if the performance of the two systems are equivalent. We are able to show that even if all the individual classifiers have been optimized, dynamic classifier selection by local accuracy is still capable of improving overall performance significantly. By contrast, simple voting techniques, and even a recently proposed CMC algorithm, were not able to show any significant improvement. It is expected that advances in the performance of low-level classification tasks via CMC will ultimately result in diagnostic accuracy at the image level that would not be possible using any single classifier.

3.4 Segmentation Techniques

We have compared two general approaches to segmenting potential lesions from digital mammogram images: contrast-based and texture-based segmentation. There would appear to be a great advantage to using texture features to key on suspicious regions in the images. A dramatic reduction of false positive regions segmented with a very small drop in sensitivity can be achieved by considering the texture of the breast tissue rather than intensity. This work is in the preliminary stages, but our direction for future work would seem clear. Namely, to develop texture-based segmentation routines for digital mammography.

References

[1] J. A. Swets, "ROC analysis applied to the evaluation of medical imaging techniques," *Investigative Radiology*, vol. 14, pp. 109–121, 1979.

[2] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a reciever operating characteristic (ROC) curve," *Radiology*, vol. 143, pp. 29–36, 1982.

[3] C. E. Metz, "ROC methodology in radiologic imaging," *Investigative Radiology*, vol. 21, pp. 720–733, 1986.

[4] D. K. McClish, "Analyzing a portion of the roc curve," *Medical Decision Making*, vol. 9, pp. 190–195, 1989.

[5] H. Drucker, R. Schapire, and P. Simard, "Boosting performance in neural networks," *Int. Journal of Pattern Recognition and Artificial Intelligence*, vol. 7, no. 4, pp. 705–719, 1993.

[6] T. K. Ho, J. J. Hull, and S. N. Srihari, "Decision combination in multiple classifier systems," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 66–75, January 1994.

[7] Y. S. Huang and C. Y. Suen, "A method of combining multiple classifiers - a neural network approach," in *Proceedings of the 12th International Conference on Pattern Recogntion and Computer Vision*, (Jerusalem, Israel), pp. 473–475, 1994.

[8] Y. S. Huang and C. Y. Suen, "A method of combining multiple experts for the recognition of unconstrained handwritten numerals," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, no. 1, pp. 90–94, 1995.

[9] F. Kimura and M. Shridar, "Handwritten numerical recognition based on multiple algorithms," *Pattern Recognition*, vol. 24, no. 10, pp. 969–983, 1991.

[10] L. Lam and C. Y. Suen, "A theoretical analysis of the application of majority voting to pattern recognition," in *Proceedings of the 12th International Conference on Pattern Recogntion and Computer Vision*, (Jerusalem, Israel), pp. 418–420, 1994.

[11] C. Nadal, R. Legault, and C. Y. Suen, "Complementary algorithms for the recognition of totally unconstrained handwritten numerals," in *Proceedings of the 10th International Conference on Pattern Recogntion*, (Atlantic City, NJ), pp. 443–449, 1990.

[12] M. Sabourin, A. Mitiche, D. Thomas, and G. Nagy, "Classifier combination for handprinted digit recognition," in *Proceedings of the 2nd International Conference on Document Analysis and Recognition*, (Tsukuba Saenie City, Japan), pp. 163–166, 20-22 Oct. 1993.

[13] L. Xu, A. Krzyzak, and C. Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 3, pp. 418–435, 1992.

[14] W. P. Kegelmeyer, Jr., “Computer detection of stellate lesions in mammograms,” in *Proceedings of SPIE Conference on Biomedical Image Processing*, 1992.

[15] W. P. Kegelmeyer, Jr., “Evaluation of stellate lesion detection in a standard mammogram data set,” in *Proceedings of the SPIE/IS&T Symposium on Electronic Imaging Science and Technology*, vol. 1905, (San Jose, CA), pp. 787–798, Jan 31 - Feb 4 1993.

[16] W. P. Kegelmeyer, Jr., J. M. Pruneda, P. D. Bourland, A. H. Hillis, M. W. Riggs, and M. L. Nipper, “Computer-aided mammographic screening for spiculated lesions,” *Radiology*, vol. 191, pp. 331–337, 1994.

[17] K. S. Woods, *Automated Image Analysis Techniques for Digital Mammography.* PhD thesis, University of South Florida, 1994.

[18] K. Knight, “Connectionist ideas and algorithms,” *Communications of the ACM*, vol. 33, no. 11, pp. 59–74, 1990.

[19] K. S. Woods, J. L. Solka, C. E. Priebe, C. C. Doss, K. W. Bowyer, and L. P. Clarke, “Comparative evaluation of pattern recognition techniques for detection of microcalcifications,” in *State of the Art in Digital Mammographic Image Analysis* (K. W. Bowyer and S. Astley, eds.), pp. 213–231, Singapore: World Scientific Publishing Company, 1994.

[20] I. N. Bankman, V. G. Sigillito, R. A. Wise, and P. L. Smith, “Feature-based detection of the K-complex wave in the human electroencephalogram using neural networks,” *IEEE Transactions on Biomedical Engineering*, vol. 39, no. 12, pp. 1305–1310, 1992.

[21] J. M. Boone, V. G. Sigillito, and G. S. Shaber, “Neural networks in radiology: An introduction and evaluation in a signal detection task,” *Medical Physics*, vol. 17, pp. 234–241, Mar/Apr 1990.

[22] G. W. Gross, J. M. Boone, V. Greco-Hunt, and B. Greenberg, “Neural networks in radiologic diagnosis: II. interpretation of neonatal chest radiographs,” *Investigative Radiology*, vol. 25, pp. 1017–1023, 1990.

[23] Y. Wu, K. Doi, M. L. Giger, and R. M. Nishikawa, “Computerized detection of clustered microcalcifications in digital mammograms: Applications of artificial neural networks,” *Medical Physics*, vol. 19, pp. 555–560, May/June 1992.

[24] J. A. Hanley and B. J. McNeil, “A method of comparing the areas under reciever operating characteristic curves derived from the same cases,” *Radiology*, vol. 148, pp. 839–843, 1983.

[25] W. P. Kegelmeyer, Jr. and M. C. Allmen, “Dense feature maps for detection of calcifications,” in *Digital Mammography: Proceedings of the 2nd International Workshop on Digital Mammography*, vol. 1069 of *International Congress Series*, (York, England), pp. 3–12, Elsevier Science B. V., July 10-12 1994.

[26] N. Karssemeijer, "Adaptive noise equalization and recognition of microcalcification clusters in mammograms," in *State of the Art in Digital Mammographic Image Analysis* (K. W. Bowyer and S. Astley, eds.), pp. 148–166, Singapore: World Scientific Publishing Company, 1994.

[27] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis.* John Wiley and Sons, 1973.